

Klasifikasi Ras Kucing Real-Time Pada Android Menggunakan Transfer Learning MobileNetV2

Mohammad Ubaidillah Ridlo¹

¹Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri
E-mail: *¹ubaidillahridlo57@gmail.com

Abstrak – Identifikasi ras kucing penting dilakukan karena setiap ras memiliki karakteristik dan kebutuhan perawatan yang berbeda, sementara masyarakat awam sering mengalami kesulitan membedakan ras kucing secara visual akibat kemiripan fitur. Penelitian ini mengembangkan sistem klasifikasi ras kucing secara real-time pada perangkat Android menggunakan arsitektur MobileNetV2 dengan pendekatan transfer learning. Dataset terdiri dari 1.000 citra yang mencakup lima ras kucing, yang kemudian melalui tahap preprocessing dan augmentasi data secara intensif untuk mencegah overfitting pada keterbatasan jumlah sampel. Model dilatih menggunakan TensorFlow dan dikonversi ke format TensorFlow Lite untuk mendukung efisiensi komputasi di perangkat mobile. Hasil pengujian menunjukkan akurasi validasi sebesar 92%, dengan performa klasifikasi sangat baik pada kelas berkarakteristik visual distingtif seperti Bombay. Implementasi pada aplikasi Android menggunakan CameraX menunjukkan bahwa proses inferensi berlangsung sangat responsif dengan rata-rata waktu deteksi 75–85 ms. Hasil ini membuktikan bahwa MobileNetV2 efektif digunakan sebagai model ringan namun akurat untuk klasifikasi citra di lingkungan perangkat bergerak.

Kata Kunci — *Android, Klasifikasi Citra, MobileNetV2, Ras Kucing, Transfer Learning*

1. PENDAHULUAN

Kucing merupakan salah satu hewan peliharaan yang paling banyak diminati di berbagai negara, termasuk Indonesia, karena tingkah lakunya yang menarik dan variasi ras yang beragam. Setiap ras kucing memiliki karakteristik fisik, temperamen, dan kebutuhan perawatan kesehatan yang spesifik. Namun, identifikasi ras kucing secara manual sering kali menjadi tantangan tersendiri, terutama bagi masyarakat awam atau pemilik pemula. Kesalahan identifikasi kerap terjadi pada ras yang memiliki kemiripan fitur visual yang tinggi, seperti pola warna bulu atau struktur wajah yang serupa. Kondisi ini menunjukkan adanya urgensi untuk mengembangkan sistem yang dapat membantu proses identifikasi secara otomatis dan akurat.

Dalam dekade terakhir, perkembangan teknologi kecerdasan buatan, khususnya *Deep Learning*, telah memberikan solusi signifikan dalam pengolahan citra digital. Studi literatur menunjukkan bahwa *Convolutional Neural Network* (CNN) merupakan metode yang paling dominan dan efektif untuk tugas klasifikasi citra dibandingkan metode konvensional [1]. Berbagai penelitian empiris telah membuktikan keandalan CNN dalam mengenali objek hewan dengan tingkat akurasi yang tinggi, baik pada kasus klasifikasi biner anjing dan kucing [2], maupun pada klasifikasi multi-kelas yang lebih kompleks seperti pembedaan spesies anjing, kucing, dan harimau [3]. Kemampuan CNN untuk mengekstraksi fitur visual secara hierarkis menjadikannya standar emas dalam visi komputer saat ini.

Meskipun memiliki performa tinggi, penerapan CNN konvensional pada perangkat bergerak (*mobile*) menghadapi kendala komputasi. Arsitektur klasik seperti ResNet50V2, yang digunakan dalam penelitian sebelumnya, memiliki jumlah parameter yang sangat besar sehingga membebani memori dan prosesor perangkat [4]. Masalah serupa juga ditemukan pada aplikasi klasifikasi ras kucing berbasis Android yang menggunakan arsitektur CNN konvensional, di mana kompleksitas model menyebabkan inefisiensi saat dijalankan pada smartphone dengan spesifikasi terbatas [5]. Kondisi ini menjadi celah penelitian untuk mencari keseimbangan antara akurasi model dan efisiensi sumber daya.

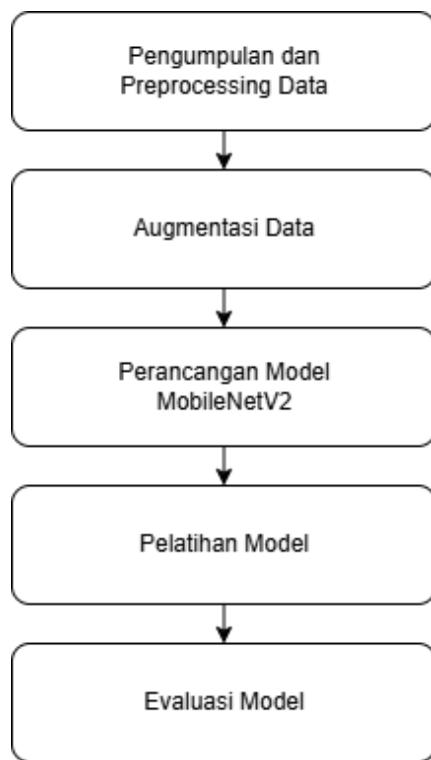
Sebagai solusi atas permasalahan tersebut, arsitektur MobileNetV2 hadir dengan pendekatan *depthwise separable convolution* yang dirancang khusus untuk mengurangi beban komputasi tanpa mengorbankan akurasi secara signifikan. Efektivitas MobileNetV2 dalam klasifikasi ras kucing telah dibuktikan pada penelitian sebelumnya dengan ukuran model yang relatif ringan dan performa yang kompetitif [6]. Selain pemilihan arsitektur, penerapan strategi *transfer learning* juga berperan penting dalam meningkatkan stabilitas model pada dataset terbatas, sebagaimana diterapkan pada kasus identifikasi tumor otak dan klasifikasi fauna laut [7], [8].

Pemanfaatan teknologi berbasis Android untuk pengelolaan dan pemantauan hewan peliharaan juga telah menjadi fokus riset dalam lingkup seminar nasional teknologi saat ini [9]. Sejalan dengan tren tersebut dan standar pengujian aplikasi berbasis visi komputer [10], penelitian ini bertujuan untuk mengembangkan sistem klasifikasi

ras kucing secara *real-time* pada platform Android. Sistem ini mengintegrasikan arsitektur MobileNetV2 dengan pendekatan *transfer learning* untuk menghasilkan aplikasi yang tidak hanya akurat, tetapi juga responsif dan ringan bagi pengguna.

2. METODE PENELITIAN

Penelitian ini menerapkan pendekatan eksperimental untuk membangun model *deep learning* yang optimal pada perangkat *mobile*. Tahapan penelitian disusun secara sistematis mulai dari pengumpulan data hingga integrasi sistem, sebagaimana diilustrasikan pada Gambar 1.



Gambar 1. Diagram Alur Tahapan Penelitian

2.1 Pengumpulan dan Preprocessing Data

Dataset yang digunakan dalam penelitian ini berjumlah total 1.000 citra digital yang terbagi secara merata ke dalam lima kelas ras kucing, yaitu: Bombay, British Shorthair, Egyptian Mau, Maine Coon, dan Persian. Setiap kelas memiliki 200 sampel citra. Struktur pembagian data ini mengacu pada standar dataset seimbang (*balanced dataset*) yang disarankan dalam penelitian klasifikasi ras kucing untuk menghindari bias prediksi pada kelas mayoritas [4], [11].

Sebelum dimasukkan ke dalam model, seluruh citra melalui tahap *preprocessing*. Citra asli yang memiliki dimensi bervariasi diubah ukurannya (*resizing*) menjadi format standar 224×224 piksel sesuai *input layer* MobileNetV2. Selanjutnya, dilakukan normalisasi nilai piksel dari rentang 0–255 menjadi 0–1 menggunakan persamaan (1):

$$x' = \frac{x}{255} \dots \dots \dots (1)$$

Normalisasi ini bertujuan untuk mempercepat konvergensi algoritma optimasi selama proses pelatihan [2].

2.2 Augmentasi Data

Mengingat keterbatasan jumlah dataset (1.000 citra), teknik augmentasi data diterapkan secara agresif untuk memperkaya variasi fitur visual dan mencegah *overfitting*. Teknik ini memanipulasi citra latih secara acak tanpa mengubah informasi labelnya. Transformasi yang diterapkan meliputi rotasi, *flipping* horizontal, *zoom*, dan pergeseran (*translation*). Proses augmentasi ini dimodelkan sebagai fungsi transformasi T pada citra input x seperti pada persamaan (2):

$$\hat{x} = T(x; \theta) \dots \dots \dots (2)$$

Dimana θ adalah parameter acak transformasi. Secara spesifik, operasi rotasi menggunakan matriks transformasi geometri sesuai persamaan (3):

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \dots\dots(3)$$

Penerapan teknik augmentasi ini telah terbukti krusial untuk meningkatkan ketahanan (*robustness*) model CNN pada kasus klasifikasi hewan dengan variasi data terbatas, sebagaimana dibuktikan dalam penelitian [2] dan [6].



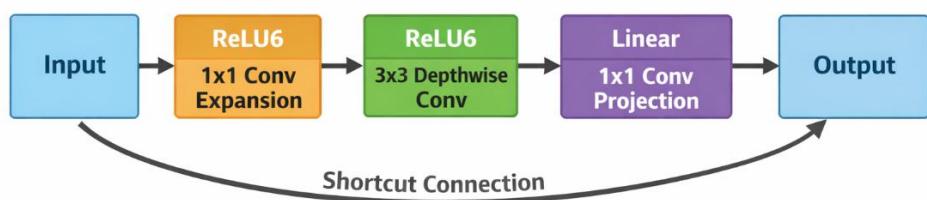
Gambar 2. Contoh Hasil Augmentasi Data: (a) Asli, (b) Rotasi, (c) Zoom, (d) Brightness

2.3 Perancangan Model MobileNetV2

Penelitian ini mengadopsi arsitektur MobileNetV2 yang dikembangkan oleh Google. Keunggulan utama arsitektur ini terletak pada penggunaan *Inverted Residual Blocks* dan *Depthwise Separable Convolution*. Berbeda dengan konvolusi standar, metode ini membagi proses konvolusi menjadi dua tahap: *depthwise convolution* untuk pemfilteran spasial ringan, diikuti oleh *pointwise convolution* (1×1) untuk kombinasi fitur. Struktur ini secara signifikan mengurangi beban komputasi (FLOPs) sehingga sangat ideal untuk perangkat Android [6], [7].

Selain itu, fungsi aktivasi yang digunakan adalah ReLU6. Fungsi ini dipilih untuk membatasi nilai aktivasi maksimal agar tetap menjaga presisi numerik pada perangkat *mobile* dengan daya rendah, sebagaimana didefinisikan secara matematis dalam persamaan (4):

Pada lapisan terakhir, *layer* klasifikasi diganti dengan *Fully Connected Layer* baru yang memiliki 5 *neuron output* sesuai jumlah kelas ras kucing, diakhiri dengan fungsi aktivasi Softmax untuk menghitung probabilitas prediksi.



Gambar 3. Struktur Inverted Residual Block pada MobileNetV2

2.4 Pelatihan Model

Pelatihan Model
Model dilatih menggunakan strategi *Transfer Learning*, di mana bobot awal model diambil dari pra-pelatihan pada dataset ImageNet. Parameter pelatihan dikonfigurasi menggunakan *optimizer* Adam dengan *learning rate* sebesar 0.0001, ukuran *batch* 32, dan pelatihan dilakukan hingga mencapai konvergensi model. Pemilihan *optimizer* Adam didasarkan pada kemampuannya beradaptasi terhadap *sparse gradients* yang sering muncul pada klasifikasi citra kompleks [6]. Fungsi kerugian (*loss function*) yang diminimalkan adalah *Categorical Cross-Entropy*, yang didefinisikan secara matematis pada persamaan (5):

$$L = - \sum_{i=1}^C y_i \cdot \log (\hat{y}_i) \dots \dots \dots (5)$$

Dimana C adalah jumlah kelas, y_i adalah label target, dan \hat{y}_i adalah probabilitas prediksi model.

2.5 Evaluasi Model

Evaluasi performa model dilakukan secara kuantitatif berdasarkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang diperoleh dari tabel *Confusion Matrix* [8]. Metrik yang digunakan meliputi Akurasi untuk mengukur ketepatan global, Presisi untuk mengukur ketepatan prediksi positif, Recall untuk mengukur rasio positif yang berhasil dideteksi, dan F1-Score sebagai rata-rata harmonis. Perhitungan matematis untuk metrik-metrik tersebut ditunjukkan pada persamaan (6) hingga (9):

Akurasi

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(6)$$

Precision

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(7)$$

Recall

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(8)$$

F1-score

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots\dots\dots(9)$$

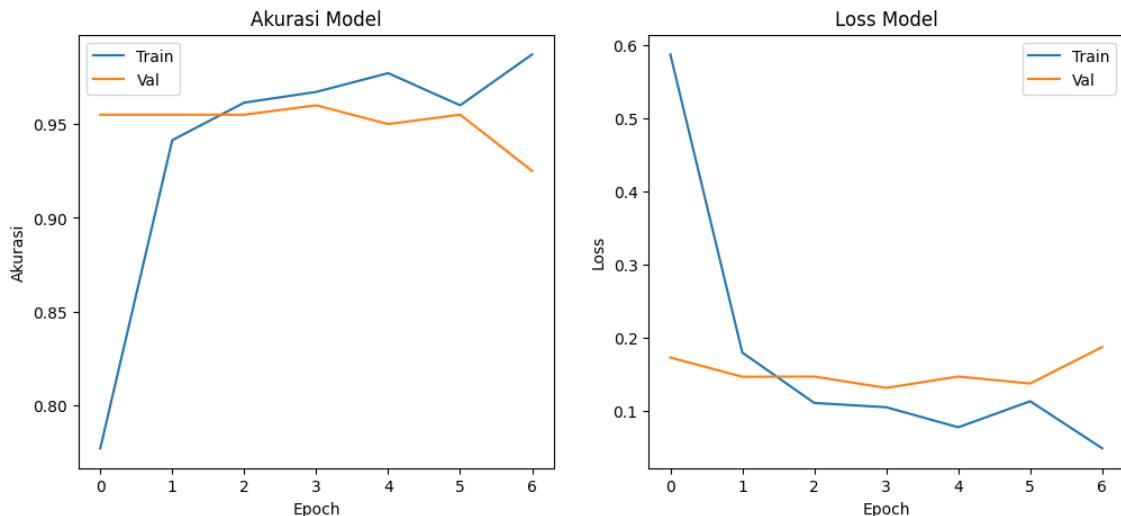
Selain evaluasi akurasi, pengujian kinerja komputasi dilakukan langsung pada perangkat Android untuk memvalidasi klaim *real-time*. Parameter kunci yang diukur adalah rata-rata waktu inferensi (*inference time*), yaitu durasi yang dibutuhkan model untuk memproses satu *frame* citra hingga menghasilkan prediksi kelas. Pengukuran latensi dalam satuan milidetik (ms) ini bertujuan untuk membuktikan efisiensi arsitektur MobileNetV2 dalam lingkungan sumber daya terbatas, sejalan dengan tren pengembangan aplikasi berbasis Android terkini [9].

3. HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil eksperimen pelatihan model, analisis performa klasifikasi, serta evaluasi implementasi sistem pada perangkat Android.

3.1 Hasil Pelatihan Model

Proses pelatihan model MobileNetV2 dilakukan dengan konfigurasi *batch size* 32. Perubahan nilai akurasi dan *loss* pada data latih dan data validasi diamati pada setiap epoch untuk mengidentifikasi indikasi *overfitting* maupun *underfitting* selama proses pelatihan.



Gambar 4. Grafik Pergerakan Akurasi dan Loss Selama Proses Pelatihan

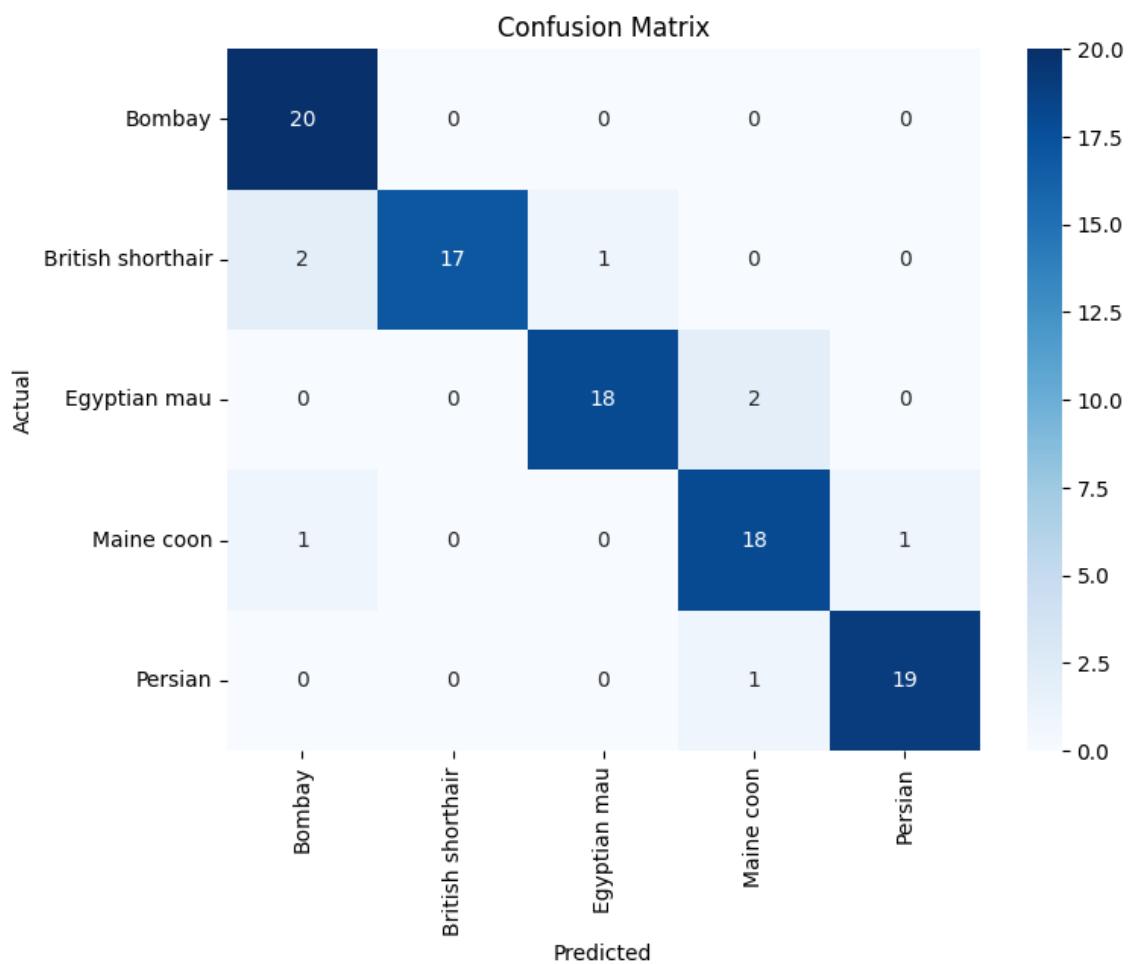
Berdasarkan Gambar 4, terlihat bahwa akurasi pelatihan mengalami peningkatan yang signifikan pada epoch awal, dari sekitar 78% hingga mendekati 98% pada epoch akhir. Sementara itu, akurasi validasi menunjukkan pola yang relatif stabil pada kisaran 92–96%, meskipun terjadi sedikit fluktuasi pada epoch terakhir. Pola ini mengindikasikan bahwa model mampu melakukan generalisasi dengan baik terhadap data yang tidak dilatih.

Pada grafik *loss*, nilai *loss* pelatihan menurun secara konsisten seiring bertambahnya epoch, sedangkan *loss* validasi cenderung stabil dengan sedikit peningkatan pada epoch terakhir. Fenomena ini menunjukkan adanya kecenderungan *overfitting ringan* pada tahap akhir pelatihan, namun masih dalam batas yang dapat diterima karena tidak diikuti dengan penurunan akurasi validasi yang drastis.

Secara keseluruhan, hasil pelatihan menunjukkan bahwa strategi augmentasi data yang diterapkan berperan dalam menjaga kestabilan performa model. Akurasi validasi terbaik yang dicapai berada pada kisaran 92%, dengan nilai *loss* validasi tetap di bawah 0,2. Capaian ini kompetitif jika dibandingkan dengan penelitian sebelumnya yang menggunakan arsitektur ResNet50V2 [4], serta menegaskan bahwa MobileNetV2 mampu memberikan performa yang tinggi dengan kompleksitas model yang lebih ringan [6].

3.2 Evaluasi Kinerja Klasifikasi

Untuk menganalisis detail kesalahan prediksi pada setiap kelas, dilakukan pengujian menggunakan *Confusion Matrix* terhadap data uji. Visualisasi hasil prediksi disajikan dalam bentuk *heatmap* pada Gambar 5.



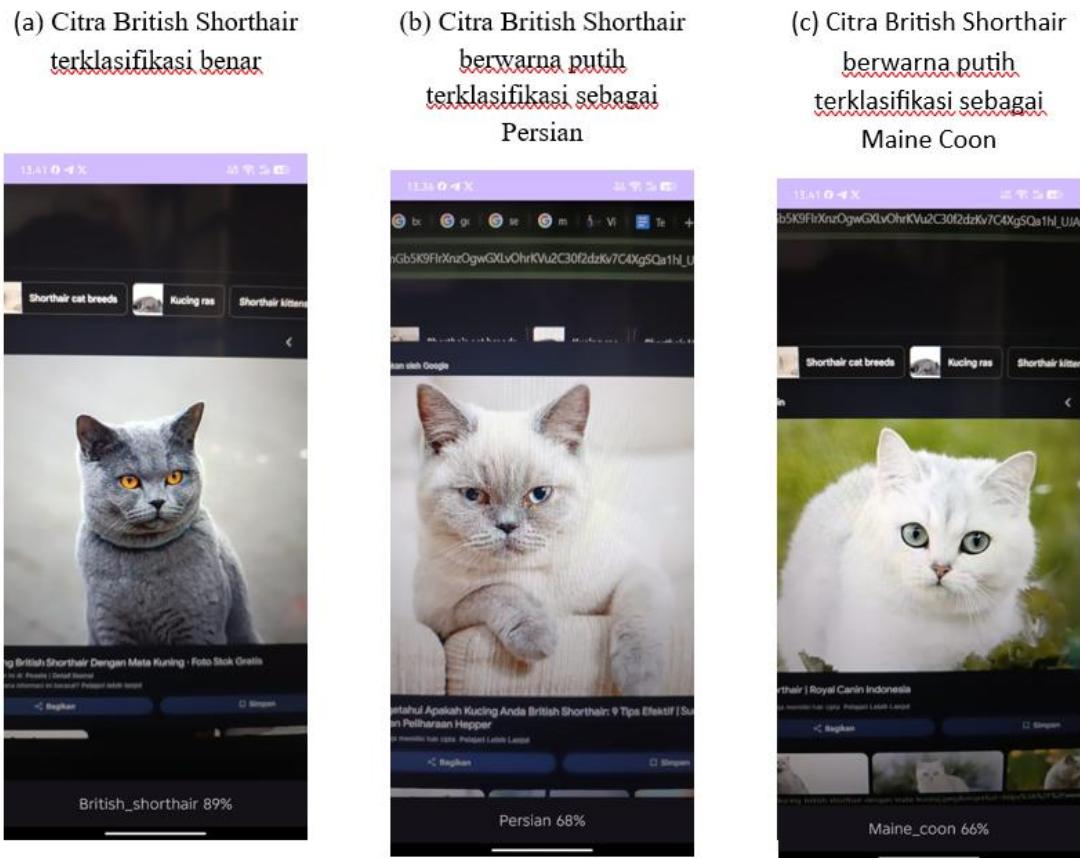
Gambar 5. Confusion Matrix Hasil Klasifikasi pada Data Uji

Berdasarkan Gambar 5, terlihat adanya variasi performa klasifikasi antar kelas. Ras Bombay menunjukkan performa terbaik dengan tingkat klasifikasi sempurna (100%). Tingginya akurasi pada kelas ini dipengaruhi oleh karakteristik visual yang sangat distingtif, yaitu warna bulu hitam solid dan kontras pigmen mata, sehingga fitur visualnya lebih mudah diekstraksi oleh lapisan konvolusi. Objek dengan karakteristik warna yang homogen dan tegas umumnya menghasilkan representasi fitur yang lebih stabil dibandingkan objek dengan tekstur dan warna yang kompleks [11].

Sebaliknya, tingkat kesalahan klasifikasi lebih sering ditemukan pada ras British Shorthair. Beberapa citra British Shorthair terklasifikasi sebagai Persian dan Maine Coon, sebagaimana tercermin pada nilai *off-diagonal* pada matriks konfusi. Fenomena ini disebabkan oleh kemiripan karakteristik morfologi antar ras, khususnya pada struktur wajah yang cenderung brachycephalic (wajah bulat), kepadatan bulu yang tinggi,

serta dominasi warna terang. Kemiripan fitur visual tersebut menyulitkan model dalam membedakan batas antar kelas, terutama ketika informasi tekstur wajah tidak cukup kontras [2], [3].

Analisis visual terhadap hasil klasifikasi ditunjukkan pada Gambar 6. Gambar 6(a) menampilkan citra British Shorthair dengan warna abu-abu yang berhasil diklasifikasikan dengan benar. Namun, pada citra British Shorthair berwarna putih, model menunjukkan kecenderungan salah klasifikasi menjadi ras Persian atau Maine Coon, sebagaimana ditunjukkan pada Gambar 6(b) dan Gambar 6(c). Hal ini mengindikasikan bahwa warna bulu yang sangat terang dan homogen dapat mendominasi proses ekstraksi fitur, sehingga karakteristik spesifik British Shorthair menjadi kurang menonjol dibandingkan ras lain dengan kemiripan visual global.



Gambar 6. Analisis Kesalahan Visual Klasifikasi British Shorthair: (a) Citra British Shorthair terklasifikasi benar, (b) Citra British Shorthair berwarna putih terklasifikasi sebagai Persian, (c) Citra British Shorthair berwarna putih terklasifikasi sebagai Maine Coon

3.3 Implementasi dan Pengujian Sistem Android

Model hasil pelatihan dikonversi ke format TensorFlow Lite (.tflite) dan diintegrasikan ke dalam aplikasi Android. Pengujian tahap akhir pada perangkat *mobile* dilakukan untuk memvalidasi dua aspek utama: keandalan fungsi fitur (*Functional Verification*) dan efisiensi waktu komputasi (*System Performance*).

Verifikasi fungsionalitas bertujuan untuk memastikan bahwa alur kerja aplikasi, mulai dari inisialisasi kamera hingga penampilan hasil prediksi, berjalan sesuai dengan rancangan antarmuka pengguna. Rangkuman hasil verifikasi fitur utama disajikan dalam Tabel 1. Berdasarkan pengujian, seluruh komponen sistem menunjukkan status "Valid", yang mengindikasikan bahwa integrasi model ke dalam lingkungan Android telah berhasil dilakukan tanpa kesalahan eksekusi (*runtime error*) [9], [10].

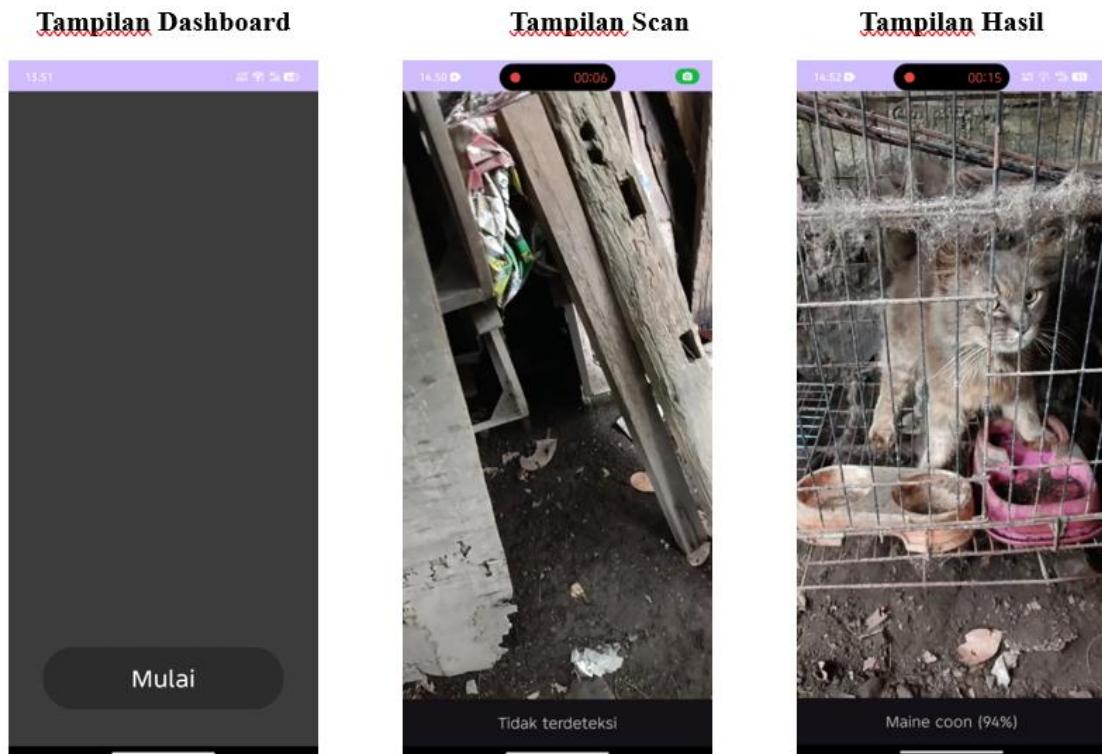
Tabel 1. Ringkasan Verifikasi Fungsionalitas Aplikasi

No	Fitur yang Diuji	Ekspektasi Sistem	Hasil Observasi	Status
1	Inisialisasi Sistem	Aplikasi terbuka dan mengakses izin kamera	Sesuai	Valid
2	<i>Image Capturing</i>	Kamera menangkap objek kucing secara <i>real-time</i>	Sesuai	Valid

3	Inferensi Model	Menampilkan nama ras dan probabilitas (%)	Sesuai	Valid
4	Fitur Informasi	Menampilkan deskripsi detail karakteristik ras	Sesuai	Valid

Selanjutnya, pengujian kinerja komputasi dilakukan untuk membuktikan klaim efisiensi waktu yang menjadi fokus utama penelitian ini. Berdasarkan uji coba langsung pada perangkat Android (Spesifikasi: RAM 8GB, perangkat Android kelas menengah), aplikasi mencatatkan rata-rata waktu inferensi (*inference time*) sebesar 75–85 ms per *frame*.

Capaian waktu ini jauh lebih cepat dibandingkan latensi pada aplikasi *Catbreedsnet* terdahulu yang masih menggunakan arsitektur CNN standar [5]. Dengan latensi di bawah 100 ms, sistem ini mampu memproses lebih dari 10 *frame* per detik (FPS), memastikan pengalaman pengguna yang mulus (*seamless*) tanpa *lag* yang berarti, sehingga layak dikategorikan sebagai sistem klasifikasi *real-time*.



Gambar 7. Tampilan Antarmuka Aplikasi Saat Deteksi Real-Time

Berdasarkan hasil pengujian dan visualisasi antarmuka pada Gambar 7, sistem yang dikembangkan tidak hanya mampu melakukan klasifikasi ras kucing secara akurat, tetapi juga dapat diimplementasikan secara efektif dalam lingkungan perangkat Android. Integrasi model MobileNetV2 dengan CameraX memungkinkan proses deteksi berjalan secara real-time dengan respons yang cepat dan antarmuka yang mudah digunakan. Temuan ini menunjukkan bahwa pendekatan yang diusulkan telah berhasil menjembatani kebutuhan akurasi model deep learning dengan keterbatasan sumber daya perangkat bergerak, sehingga sistem layak digunakan sebagai aplikasi klasifikasi ras kucing berbasis Android.

4. SIMPULAN

Berdasarkan hasil implementasi dan analisis pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Penerapan arsitektur MobileNetV2 dengan teknik *transfer learning* terbukti efektif untuk klasifikasi ras kucing pada dataset terbatas. Model mampu mencapai akurasi validasi sebesar 92,00% dan menunjukkan ketahanan yang baik terhadap *overfitting* berkat strategi augmentasi data.

2. Implementasi pada perangkat Android berhasil merealisasikan sistem klasifikasi *real-time*. Rata-rata waktu inferensi tercatat stabil pada kisaran 75–85 ms per *frame*, yang jauh lebih efisien dibandingkan arsitektur CNN standar pada penelitian terdahulu.
3. Karakteristik visual yang distingatif sangat memengaruhi performa model. Ras Bombay dengan warna solid memiliki tingkat keberhasilan deteksi tertinggi (100%), sedangkan ras British Shorthair dan Persian memiliki tingkat kesalahan prediksi yang lebih tinggi akibat kemiripan struktur morfologi wajah (*brachycephalic*).

5. SARAN

Penelitian ini masih memiliki keterbatasan yang dapat dikembangkan pada penelitian selanjutnya:

1. Perluasan dataset dengan menambahkan variasi citra pada ras yang memiliki kemiripan fisik tinggi (seperti British Shorthair dan Persian) untuk mempertegas fitur pembeda antar kelas.
2. Penerapan teknik *Fine-Tuning* pada lapisan konvolusi yang lebih dalam, bukan hanya pada *classifier*, untuk memungkinkan model mempelajari fitur-fitur spesifik ras kucing yang lebih detail.
3. Penelitian selanjutnya dapat mengkaji secara empiris pengaruh integrasi modul edukasi berbasis hasil klasifikasi ras terhadap peningkatan pemahaman pengguna melalui pengujian eksperimental.

DAFTAR PUSTAKA

- [1] K. D. Linda, Kusrini, dan A. D. Hartanto, “Studi Literatur Mengenai Klasifikasi Citra Kucing Dengan Menggunakan Deep Learning: Convolutional Neural Network (CNN),” *Journal of Electrical Engineering and Computer (JEECOM)*, vol. 6, no. 1, hal. 129–137, 2024.
- [2] T. Adriyanto, R. A. Ramadhani, R. Helilintar, dan A. Risktyawan, “Classification of Dog and Cat Images Using the CNN Method,” *ILKOM Jurnal Ilmiah*, vol. 14, no. 3, hal. 203–208, 2022.
- [3] M. Murdifin dan S. Uyun, “Klasifikasi Hewan Anjing, Kucing, dan Harimau Menggunakan Metode Convolutional Neural Network (CNN),” *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 10, no. 3, hal. 331–340, 2025.
- [4] C. Agusniar dan D. Adelia, “Penerapan Convolutional Neural Network pada Klasifikasi Jenis Ras Kucing Menggunakan ResNet50V2,” *Journal of Information Engineering and Educational Technology (JIEET)*, vol. 8, no. 1, hal. 52–60, 2024.
- [5] A. T. Ramadhan dan A. Setiawan, “Catbreedsnet: An Android Application for Cat Breed Classification Using Convolutional Neural Networks,” *JOIN (Jurnal Online Informatika)*, vol. 8, no. 1, hal. 52–60, 2023.
- [6] R. A. Hermawan, I. Taufik, dan Y. A. Gerhana, “Klasifikasi Citra Ras Kucing Berbasis CNN dengan Metode MobileNet-V2,” *INTERNAL (Information System Journal)*, vol. 8, no. 1, hal. 70–84, 2025.
- [7] M. G. Somoal dan A. R. Dzikrillah, “Komparasi MobileNetV2 dengan Kustomisasi Transfer Learning dan Hyperparameter untuk Identifikasi Tumor Otak,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 12, no. 1, hal. 229–240, 2025.
- [8] I. Fawwaz, Yennimar, N. P. Dharsinni, dan B. A. Wijaya, “The Optimization of CNN Algorithm Using Transfer Learning for Marine Fauna Classification,” *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 7, no. 4, hal. 2236–2245, 2023.
- [9] R. D. Apriansa, A. B. Cahyono, dan S. W. Hati, “Aplikasi Pengaduan Hewan Peliharaan Kota Kediri Berbasis Android,” *Prosiding Seminar Nasional Teknologi dan Sains (STAINS)*, vol. 1, no. 1, hal. 210–216, 2022.
- [10] Y. P. Astuti, E. R. Subhiyakto, I. Wardatunizza, dan E. Kartikadarma, “Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Jenis Tanah Berbasis Android,” *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 8, no. 3, hal. 220–225, 2023.
- [11] M. R. Ramadhan dan L. Wati, “Klasifikasi Ras Kucing Menggunakan Algoritma Convolutional Neural Network,” *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 9, no. 3, hal. 851–860, 2025.